TO90TT*08858660

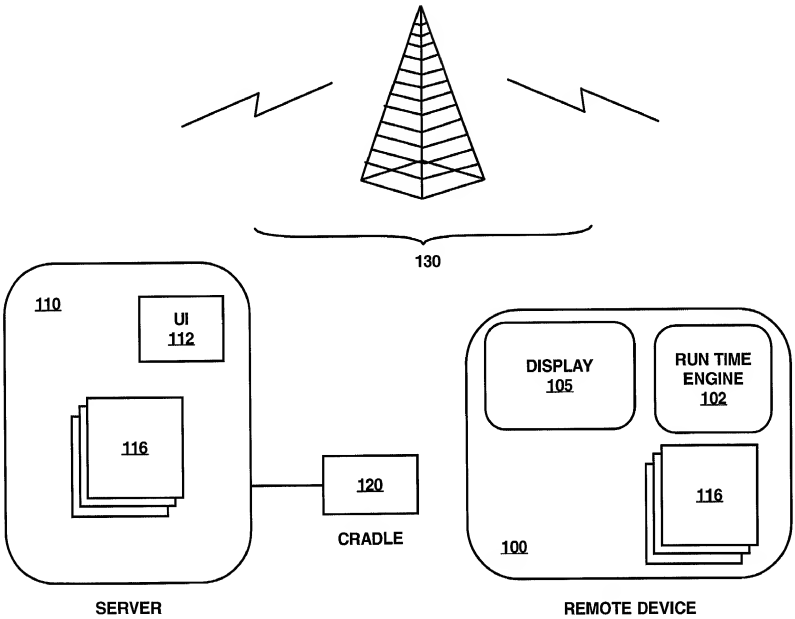

FIG 1

TD901T* 08858660

CLASS HIERARCHY ITEMS IN *ITALICS* ARE PART OF THE RIM SDK

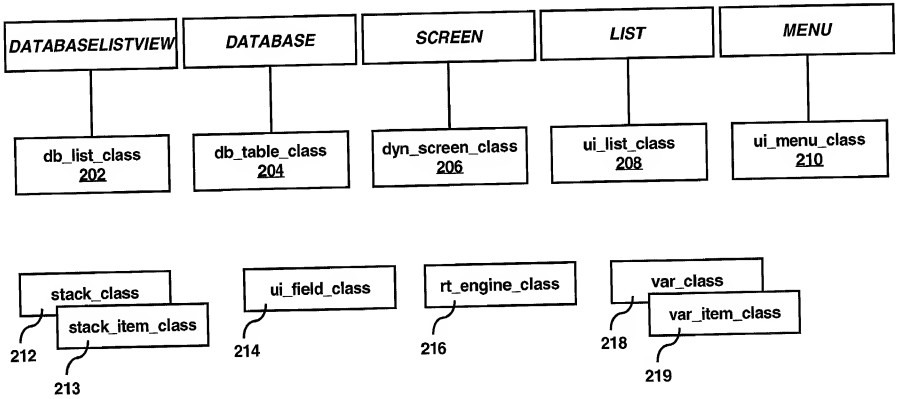

FIG 2

OBJECT RELATIONSHIPS

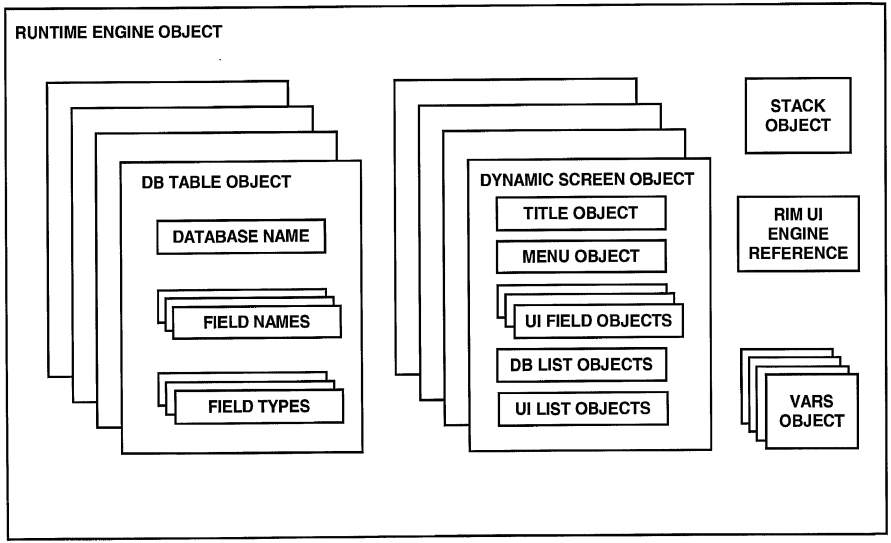

FIG 3

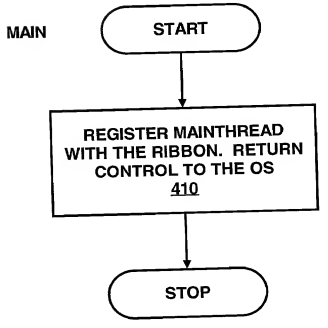
MAIN
START
REGISTER MAINTHREAD WITH THE RIBBON. RETURN CONTROL TO THE OS
410
STOP

FIG 4

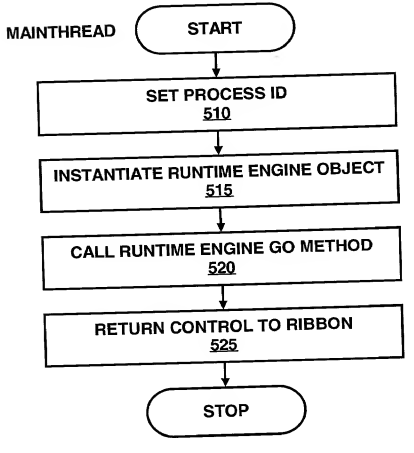
MAINTHREAD
START
SET PROCESS ID
510
INSTANTIATE RUNTIME ENGINE OBJECT
515
CALL RUNTIME ENGINE GO METHOD
520
RETURN CONTROL TO RIBBON
525
STOP

FIG 5

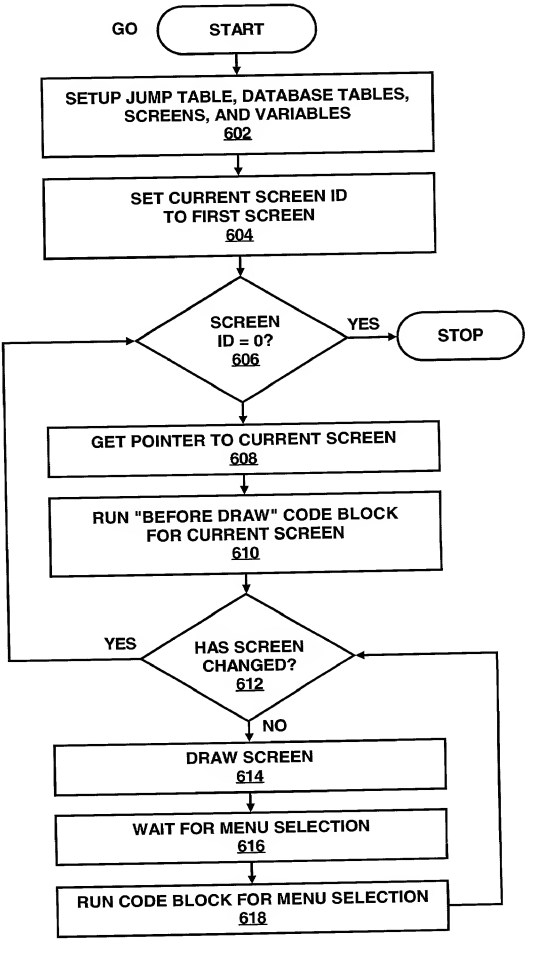
GO
START
SETUP JUMP TABLE, DATABASE TABLES, SCREENS, AND VARIABLES
602
SET CURRENT SCREEN ID TO FIRST SCREEN
604
SCREEN ID = 0?
606
YES
STOP
GET POINTER TO CURRENT SCREEN
608
RUN "BEFORE DRAW" CODE BLOCK FOR CURRENT SCREEN
610
HAS SCREEN CHANGED?
612
YES
NO
DRAW SCREEN
614
WAIT FOR MENU SELECTION
616
RUN CODE BLOCK FOR MENU SELECTION
618

FIG 6

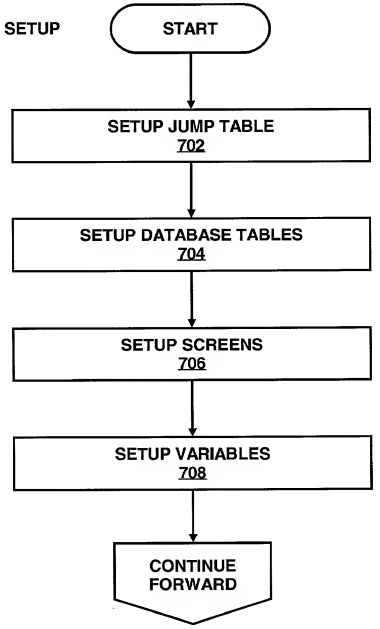
SETUP
START
SETUP JUMP TABLE
702
SETUP DATABASE TABLES
704
SETUP SCREENS
706
SETUP VARIABLES
708
CONTINUE
FORWARD

FIG 7

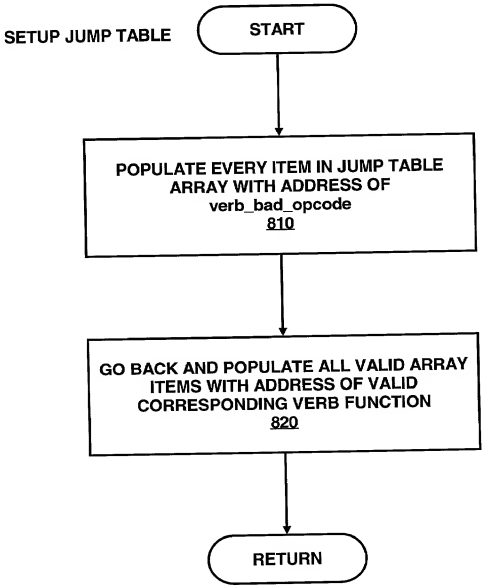
SETUP JUMP TABLE
START
POPULATE EVERY ITEM IN JUMP TABLE ARRAY WITH ADDRESS OF verb_bad_opcode
810
GO BACK AND POPULATE ALL VALID ARRAY ITEMS WITH ADDRESS OF VALID CORRESPONDING VERB FUNCTION
820
RETURN

FIG 8

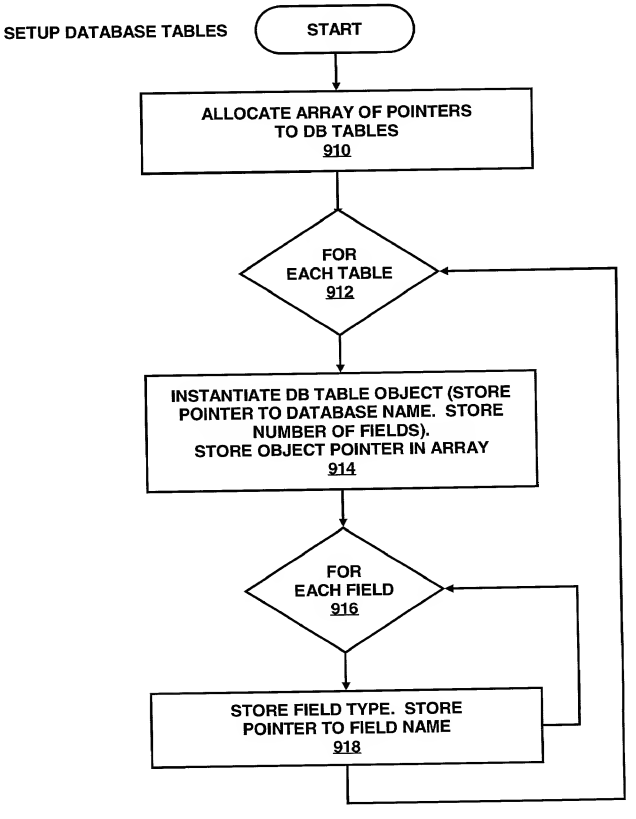
SETUP DATABASE TABLES
START
ALLOCATE ARRAY OF POINTERS TO DB TABLES
910
FOR EACH TABLE
912
INSTANTIATE DB TABLE OBJECT (STORE POINTER TO DATABASE NAME. STORE NUMBER OF FIELDS). STORE OBJECT POINTER IN ARRAY
914
FOR EACH FIELD
916
STORE FIELD TYPE. STORE POINTER TO FIELD NAME
918

FIG 9

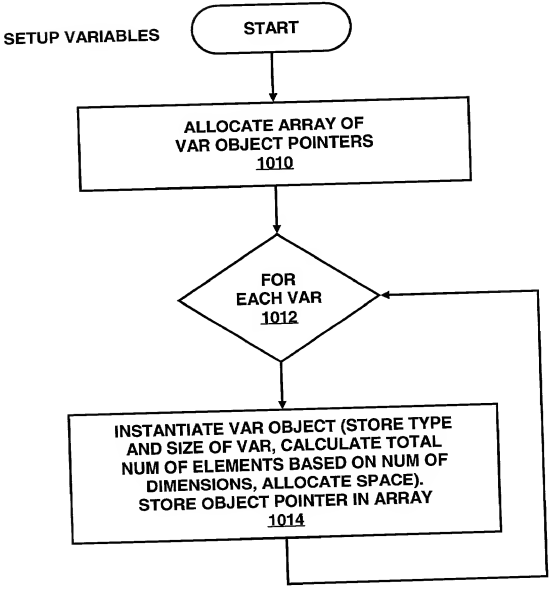
SETUP VARIABLES
START
ALLOCATE ARRAY OF VAR OBJECT POINTERS
1010
FOR EACH VAR
1012
INSTANTIATE VAR OBJECT (STORE TYPE AND SIZE OF VAR, CALCULATE TOTAL NUM OF ELEMENTS BASED ON NUM OF DIMENSIONS, ALLOCATE SPACE). STORE OBJECT POINTER IN ARRAY
1014

FIG 10

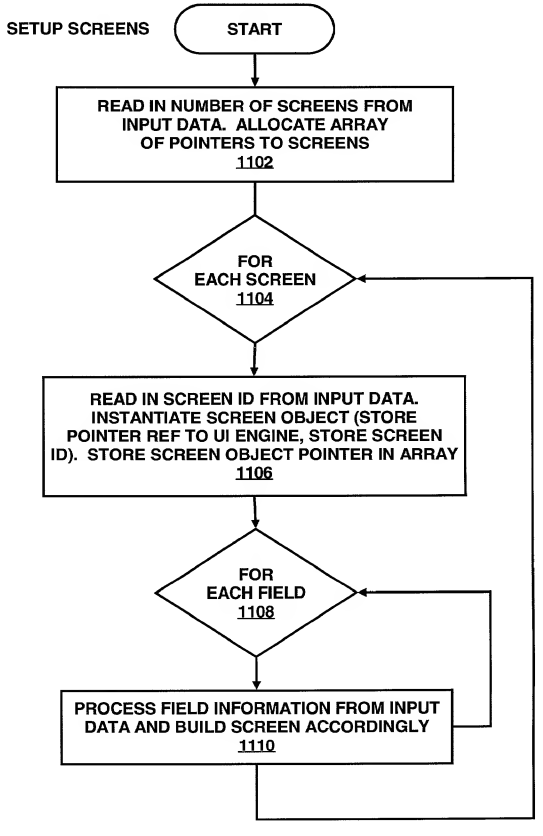
SETUP SCREENS
START
READ IN NUMBER OF SCREENS FROM INPUT DATA. ALLOCATE ARRAY OF POINTERS TO SCREENS
1102
FOR EACH SCREEN
1104
READ IN SCREEN ID FROM INPUT DATA. INSTANTIATE SCREEN OBJECT (STORE POINTER REF TO UI ENGINE, STORE SCREEN ID). STORE SCREEN OBJECT POINTER IN ARRAY
1106
FOR EACH FIELD
1108
PROCESS FIELD INFORMATION FROM INPUT DATA AND BUILD SCREEN ACCORDINGLY
1110

FIG 11

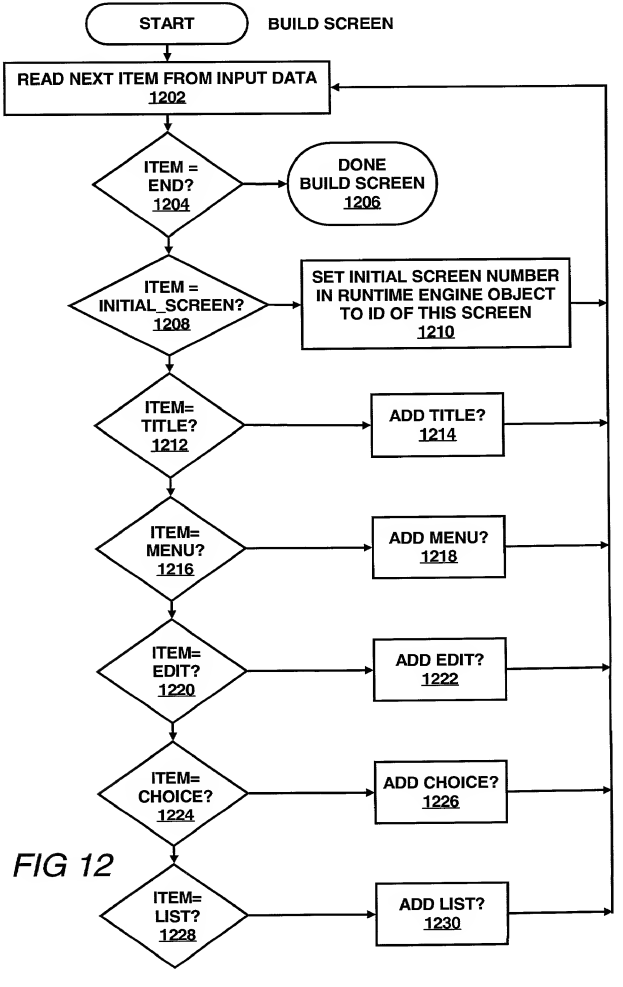
START
BUILD SCREEN
READ NEXT ITEM FROM INPUT DATA
1202
ITEM = END?
1204
DONE BUILD SCREEN
1206
ITEM = INITIAL_SCREEN?
1208
SET INITIAL SCREEN NUMBER IN RUNTIME ENGINE OBJECT TO ID OF THIS SCREEN
1210
ITEM= TITLE?
1212
ADD TITLE?
1214
ITEM= MENU?
1216
ADD MENU?
1218
ITEM= EDIT?
1220
ADD EDIT?
1222
ITEM= CHOICE?
1224
ADD CHOICE?
1226
ITEM= LIST?
1228
ADD LIST?
1230

FIG 12

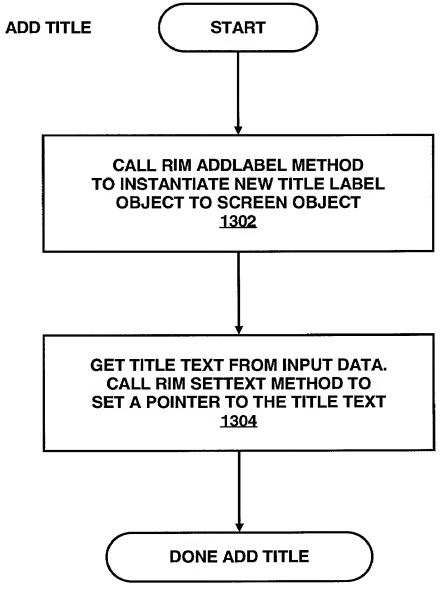
ADD TITLE
START
CALL RIM ADDLABEL METHOD TO INSTANTIATE NEW TITLE LABEL OBJECT TO SCREEN OBJECT
1302
GET TITLE TEXT FROM INPUT DATA. CALL RIM SETTEXT METHOD TO SET A POINTER TO THE TITLE TEXT
1304
DONE ADD TITLE

FIG 13

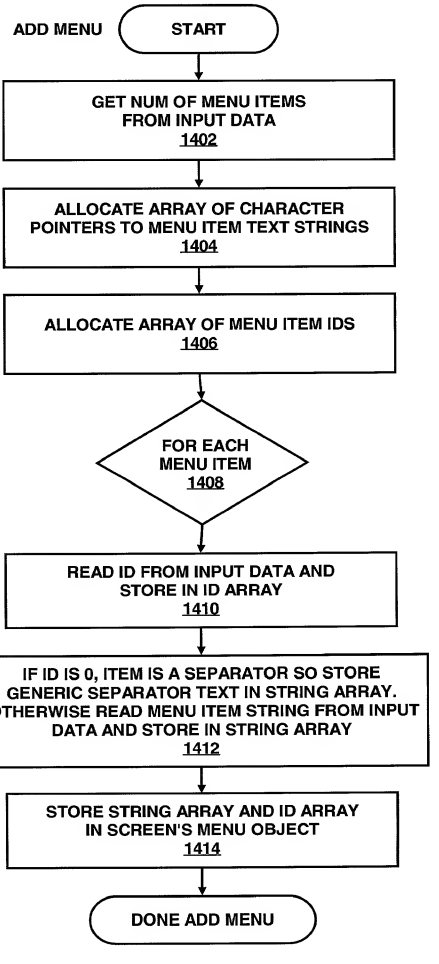
ADD MENU
START
GET NUM OF MENU ITEMS FROM INPUT DATA
1402
ALLOCATE ARRAY OF CHARACTER POINTERS TO MENU ITEM TEXT STRINGS
1404
ALLOCATE ARRAY OF MENU ITEM IDS
1406
FOR EACH MENU ITEM
1408
READ ID FROM INPUT DATA AND STORE IN ID ARRAY
1410
IF ID IS 0, ITEM IS A SEPARATOR SO STORE GENERIC SEPARATOR TEXT IN STRING ARRAY. OTHERWISE READ MENU ITEM STRING FROM INPUT DATA AND STORE IN STRING ARRAY
1412
STORE STRING ARRAY AND ID ARRAY IN SCREEN'S MENU OBJECT
1414
DONE ADD MENU

FIG 14

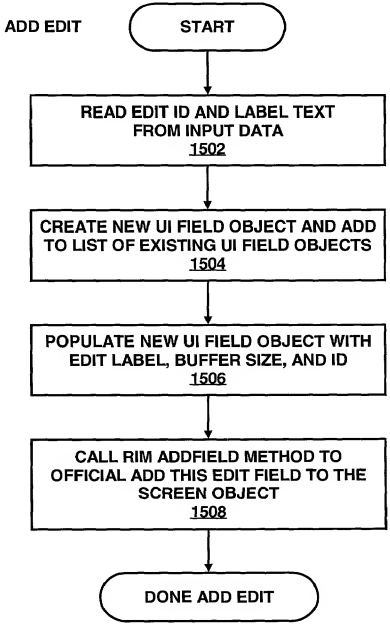

FIG 15

EDGE-US-00025660

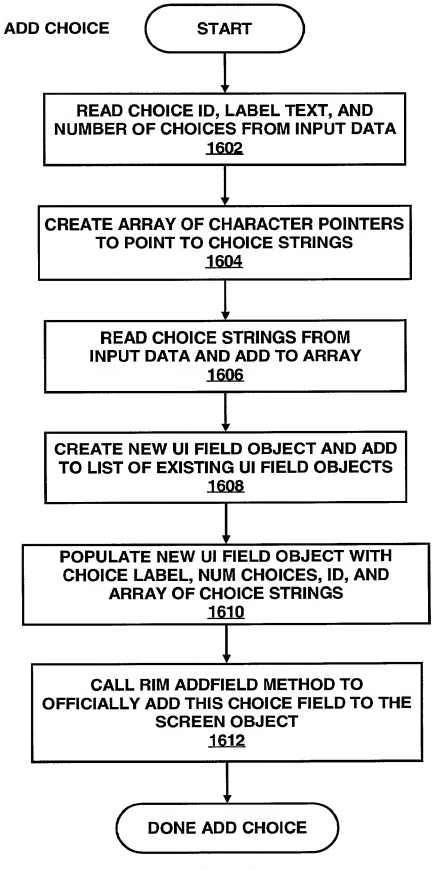
ADD CHOICE
START
READ CHOICE ID, LABEL TEXT, AND NUMBER OF CHOICES FROM INPUT DATA
1602
CREATE ARRAY OF CHARACTER POINTERS TO POINT TO CHOICE STRINGS
1604
READ CHOICE STRINGS FROM INPUT DATA AND ADD TO ARRAY
1606
CREATE NEW UI FIELD OBJECT AND ADD TO LIST OF EXISTING UI FIELD OBJECTS
1608
POPULATE NEW UI FIELD OBJECT WITH CHOICE LABEL, NUM CHOICES, ID, AND ARRAY OF CHOICE STRINGS
1610
CALL RIM ADDFIELD METHOD TO OFFICIALLY ADD THIS CHOICE FIELD TO THE SCREEN OBJECT
1612
DONE ADD CHOICE

FIG 16

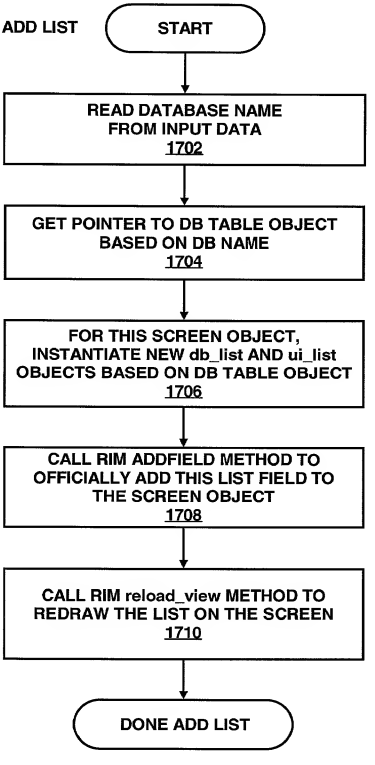
ADD LIST
START
READ DATABASE NAME FROM INPUT DATA
1702
GET POINTER TO DB TABLE OBJECT BASED ON DB NAME
1704
FOR THIS SCREEN OBJECT, INSTANTIATE NEW db_list AND ui_list OBJECTS BASED ON DB TABLE OBJECT
1706
CALL RIM ADDFIELD METHOD TO OFFICIALLY ADD THIS LIST FIELD TO THE SCREEN OBJECT
1708
CALL RIM reload_view METHOD TO REDRAW THE LIST ON THE SCREEN
1710
DONE ADD LIST

FIG 17

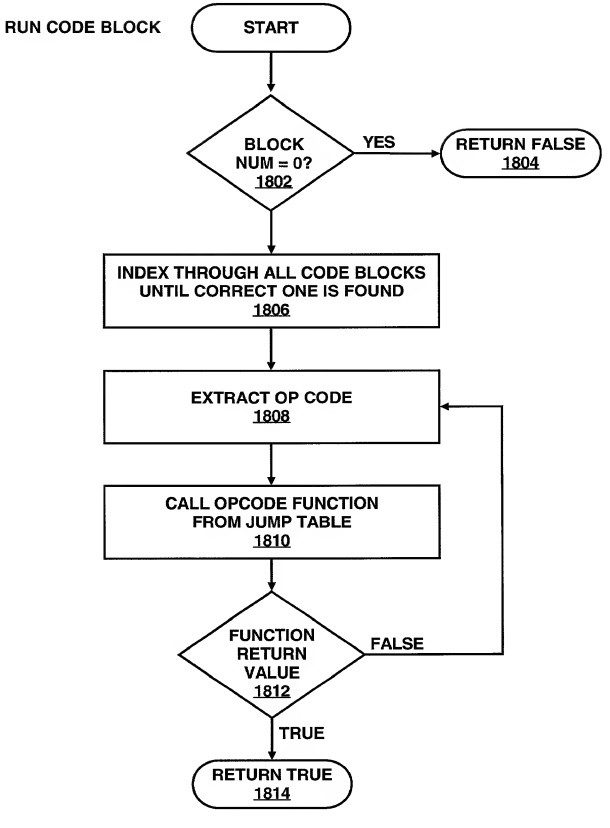
RUN CODE BLOCK
START
BLOCK NUM = 0? 1802
YES
RETURN FALSE 1804
INDEX THROUGH ALL CODE BLOCKS UNTIL CORRECT ONE IS FOUND 1806
EXTRACT OP CODE 1808
CALL OPCODE FUNCTION FROM JUMP TABLE 1810
FUNCTION RETURN VALUE 1812
FALSE
TRUE
RETURN TRUE 1814

FIG 18

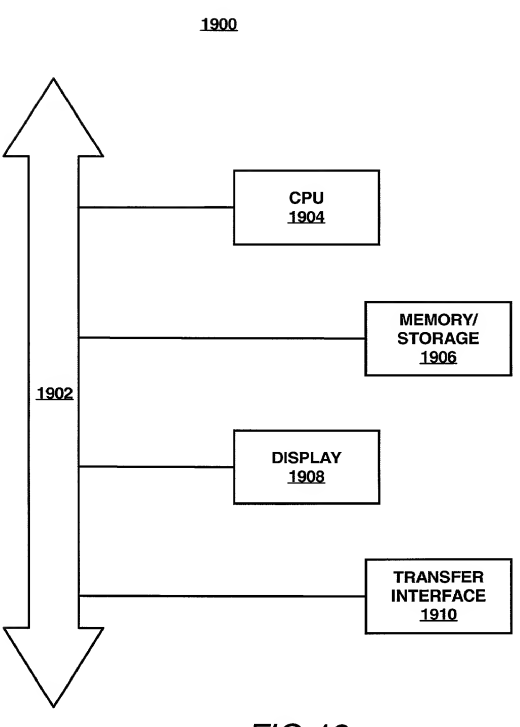
1900
CPU
1904
MEMORY/
STORAGE
1906
1902
DISPLAY
1908
TRANSFER
INTERFACE
1910

FIG 19